

acontis technologies GmbH

SOFTWARE

EC-Simulator

Python Programming Interface

Version 3.2

Edition: May 6, 2024

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

© Copyright acontis technologies GmbH

Neither this document nor excerpts therefrom may be reproduced, transmitted, or conveyed to third parties by any means whatever without the express permission of the publisher. At the time of publication, the functions described in this document and those implemented in the corresponding hardware and/or software were carefully verified; nonetheless, for technical reasons, it cannot be guaranteed that no discrepancies exist. This document will be regularly examined so that corrections can be made in subsequent editions. Note: Although a product may include undocumented features, such features are not considered to be part of the product, and their functionality is therefore not subject to any form of support or guarantee.



Contents

1	Introduction	4
	1.1 Requirements	4
	1.2 Architecture	4
2	Programmers Guide	6
	2.1 Sample Scripts	6
	2.2 Sample Code	6
	2.3 Wrapper	6
	2.3.1 Modules	6
	2.3.2 Return code vs. exception handling	7
	2.3.3 API with "out" or "ref" parameters	7
	2.4 Supported IDEs	7
	2.4.1 Python Shell IDLE	7
	2.4.2 Visual Studio 2019	9
	2.4.3 Visual Studio Code	13
3	FAQ	17



1 Introduction

The Python Wrapper provides a Python interface to use EC-Master, EC-Simulator and RAS Client/Server.

1.1 Requirements

Python v3.7 and above

- Python Pause. Required for ticked timing with pause.until(...) to lower JobTask's drift, e.g. for Distributed Clocks
 - \$ pip install pause
- PyQt5 (v5.15.1). Only required to run the GUI demo

\$ pip install pyqt5

Windows (x86/x64)

- Microsoft Windows 7 and above
- Microsoft Visual C++ 2010 Runtime

Linux (x86/x64/ARM)

• Ubuntu 12.04 and above

1.2 Architecture



The architecture contains 4 basic layers:

Customer Python Script or our examples (EcMasterDemoPython, ...)

• Demo application, written in Python

Programming Interface (EcWrapperPython)



• Provides an object oriented API written in Python

Wrapper Library (EcWrapper)

• Native wrapper library, which provides API for object oriented access

Native Libraries

- Master Core Library
- Simulator Library
- RAS Client Library



2 Programmers Guide

2.1 Sample Scripts

There are currently 2 scripts available:

```
EcMasterDemoPython.bat
Starts the console demo application
```

EcMasterDemoPythonInteractive.bat Starts the interactive demo application

The scripts will start the demo application. The interactive demo application waits for user input where the user can enter the following commands:

```
# Write variable
demo.processImage.variables.Slave_1005__EL2008_.Channel_1.Output.set(1)
# Read variable
demo.processImage.variables.Slave_1005__EL2008_.Channel_1.Output.get()
# Print properties of variable
demo.processImage.variables.Slave_1005__EL2008_.Channel_1.Output.dmp()
# Stop the demo:
demo.stopDemo()
```

2.2 Sample Code

The Python demo application contains of 3 modules:

```
EcDemoApp . py:
Console demo application
```

```
EcDemoAppGui.py:
Gui demo application, based on Qt5
```

EcDemoAppInteractive.py: Interactive demo application

2.3 Wrapper

2.3.1 Modules

The Python Wrapper contains of 4 modules:

EcWrapperPython.py

class CEcWrapperPython EC-Wrapper base class

class CEcMasterPython EC-Master

class CEcMasterMbxGatewayClientPython Mailbox Gateway Client for EC-Master



class CEcMasterMbxGatewayServerPython Mailbox Gateway Server for EC-Master

class CEcSimulatorPython EC-Simulator

class CEcSimulatorRasServerPython RAS Server for EC-Simulator

class CEcRasClientPython RAS Client for EcMaster / EcSimulator

EcWrapperPythonTypes.py

Python types

EcWrapper.py

CPython interface (internal)

EcWrapperTypes.py

CPython types (internal)

2.3.2 Return code vs. exception handling

The most of all API functions returns a return code for error handling. This behaviour can be changed to throw an exception in error case by simply setting:

CEcWrapperPython.EnableExceptionHandling = **True** # default is False

2.3.3 API with "out" or "ref" parameters

The Python Wrapper API is based on C# code. C# supports out and ref keywords for parameters. This is not supported in Python and is solved by simply submitting CEcWrapperPythonOutParam or CEcWrapper-PythonRefParam to those functions:

```
# This function has an "out" parameter "out_oSbStatus"
def GetScanBusStatus(self, out_oSbStatus):
    # ...
    return
# Create "out" parameter
out_oStatus = CEcWrapperPythonOutParam()
# Call function
pythonWrapper.GetScanBusStatus(out_oStatus)
# Get the "out" parameter value
oStatus = out_oStatus.value
# Now, the "oStatus" object can be used
print(oStatus.dwResultCode)
```

2.4 Supported IDEs

2.4.1 Python Shell IDLE

This is the default IDE.

It can be started from Windows Start Menu or by calling C:/Python/Lib/idlelib/idle.py:



🕞 Python 3.7.8 Shell —		×
<u>F</u> ile <u>E</u> dit She <u>l</u> l <u>D</u> ebug <u>O</u> ptions <u>W</u> indow <u>H</u> elp		
<pre>Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)] or Type "help", "copyright", "credits" or "license()" for more information. >>> </pre>	n win32	^
		\sim
	Ln: 3	Col: 4

In this shell, the user can simply copy&paste the sample code from: Examples/EcMasterDemoPython/EcDemoAppInteractive.py

```
exec("""
import os
import sys
INSTALLDIR = "C:/Program
Files/acontis_technologies/EC-Master-Windows-x86_64Bit/"
os.environ["PATH"] += os.pathsep + INSTALLDIR + "Bin/Windows/x64"
sys.path.append(INSTALLDIR + "Sources/EcWrapperPython")
sys.path.append(INSTALLDIR + "Examples/EcMasterDemoPython")
from EcDemoApp import \*
demo = EcMasterDemoPython()
demo.pAppParms.tRunMode = RunMode.Master
demo.pAppParms.dwBusCycleTimeUsec = 4000
demo.pAppParms.szENIFilename = "ENI.xml"
demo.pAppParms.szLinkLayer = "winpcap 127.0.0.0 1"
demo.pAppParms.nVerbose = 3
demo.startDemo()
print("EcMasterDemoPython is running.")
print("Type demo.help() for interactive help.")
""")
```

... and the demo is running.



🕞 Python 3.7.8 Shell —		×
<u>F</u> ile <u>E</u> dit She <u>l</u> l <u>D</u> ebug <u>O</u> ptions <u>W</u> indow <u>H</u> elp		
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)]	on win3	2 ^
Type "help", "copyright", "credits" or "license()" for more information.		
>>> exec("""		
import os		
import sys		
INSTALLDIR = "C:/Temp/EC-Master-Windows-x86_64Bit/"		
os.environ["PATH"] += os.pathsep + INSTALLDIR + "Bin/Windows/x64"		
sys.path.append(INSTALLDIR + "Sources/EcWrapperPython")		
sys.path.append(INSTALLDIR + "Examples/EcMasterDemoPython")		
from EcDemoApp import *		
demo = EcMasterDemoPython()		
demo.pAppParms.tRunMode = RunMode.Master		
demo.pAppParms.dwBusCycleTimeUsec = 4000		
<pre>demo.pAppParms.szENIFilename = "d:/project.xml"</pre>		
demo.pAppParms.szLinkLayer = "winpcap 172.20.143.181 1"		
demo.pAppParms.nVerbose = 1		
demo.startDemo()		
print("EcMasterDemoPython is running.")		
print("Type demo.help() for interactive help.")		
^{"""})		
EtherCAT network adapter MAC: 64-70-02-04-D9-A3		
EchasterDemoDurbon is running		
Turne demo help() for interactive belo		
The demonstrate () for interactive help.		
		~
	Ln: 25	Col: 4

2.4.2 Visual Studio 2019

Create a new project:

proiect	<u>S</u> earch f	or templates (Alt+S)			÷ 0		<u>C</u> lear a
	Python		• All <u>p</u> latf	orms	-	All project <u>t</u> ypes	
Recent project templates							
RJ Python Application Python	Z,	Python Application A project for creatin Python Windo	g a command- ws Linux	line applica macOS	tion Console		
御 Dynamic-Link Library (DLL) C++	H Y	Web Project A project for creatin	g a generic Pyti	hon web pr	oject		
🞬 Console App C++		Python Windo	ws Linux	macOS	Web		
	()	Django Web Project A project for creatin sample pages that u Python Windo	g an application se the Twitter B ws Linux	n using the ootstrap fra macOS	Django web f imework for r Web	ramework. It features esponsive web desigr	h.
	H	Flask Web Project A project for creatin template engine. It f for responsive web o	g an application eatures sample lesign.	n using the pages that	Flask web fra use the Twitte	mework with the Jinja er Bootstrap framewo	rk
		Python Windo	ws Linux	macOS	Web		



Configure the project:

• Replace the generated file EcMasterDemoPython.py with the existing EcDemoApp.py.

Configure your new project						
Python Application Python Windows Linux macOS Console						
Project <u>n</u> ame						
EcMasterDemoPython						
Location						
C:\Temp\EC-Master-Windows-x86_64Bit\Examples\	•	<u>.</u>				
Solution na <u>m</u> e 🕕 EcMasterDemoPython						
Place solution and project in the same directory						
			Back	Cre	ate	1
			Dack		ale	

Configure project *General* settings:

• Startup File: EcDemoApp.py



🔀 Eile Edit View Git Project Build Debug Test	Analyze Tools Ext	nsions Window Help Search (Ctrl=C) 🔎 EcMasterDemoPython	🐠 – 🕫 🛛
🍈 🗢 💿 🛛 🕶 🔛 🔐 🦃 🤊 - 🖓 - Debug 🕒 Any CP	VU	- [#] @ _ ※ 告 师 当 知 異 知 知 .	🔄 Live Share 🖉
Python 3.7 (64-bit)			
Solution Explorer * 9 ×	EcMasterDemoPython	• X	≁ ¢ ξ
00430-5262/-	General		VerE
Search Solution Explorer (Ctrl+ii)	Debug	nfiguration: N/A V Platform N/A V	pl.
Solution 'EcMasterDemoPython' (1 of 1 project)	Publish		2
EcMasterDemoPython	Test	Application	
 References 		Startup File (EGD:moApp.py)	*
▶ ■■ Search Paths		Working Directory	Notif
РТ Еслетолрр.ру		Windows Application	at.
		Interpreter: (Use global default)	× 8.
			Prog
			ertie.
			Ĩ.
	_		
	Output		+ 0 ×
	Show output from:	· 2 2 2 2 2 2 2	
Solution Ex Property M Class View Team Explo Test Explorer	Developer PowerShell	Output Find Symbol Results Error List	
CT Ready			Add to Source Control 🔹 🦾
L (100)			

Configure project *Debug* settings:

• Search Paths:

//Sources/EcWrapperPython;/EcMasterDemoPython							
• Script Arguments:							
mode 1 -f ENI.xmllink "winpcap 127.0.0.0 1 1" -b 4000 -t 1000 -v 3							
Environment Variables:							
PATH=//Bin/Windows/x64;%PATH%							



🕼 Elek Edet Yew Git Degiest Ruids Debug Test Agalyze Jools Extensions Window Help Search (Cat+C) 🔑 Ecklaster/Demolython						
◎ ● ◎ 初 • ▲ 単 単 ⑦ • ♡ + ♡ - ● bebug • Any CPU • ▶ Stat • • 月 回 通知時 日本 月 常常有量						
Python 3.7 (64-bit) • 🗰 🖬 🛫						
Solution Explorer * 4 ×	EcMasterDemoPytho	on* e X		- * š		
○ ○ 公 詞 ◎ - ち お 御 御 チー Search Solution Explorer (Ctrl+辺) ター	General Debug*	Configuration: N/A	V Platform N/A V	ver Explo		
Solution 'EcMasterDemoPython' (1 of 1 project)	Publish			²		
EcMasterDemoPython Det Python Environments	Test	Launch mode: Standard P	thon launcher	✓ 8		
•• References		Run		×		
Fill Search Paths Fr EcDemoAnn.nv		Search Paths	Sources'EdWapperPrine:			
		Script Arouments:		atio .		
		Interpreter Path				
		Interpreter Arguments				
		Environment Variabler	DETER 1 VIBrit Mondeaux VIL-PS 001746	Ē		
		chinionment ganables.	Lan La - Print Cantin Const York Schol Like			
		Debug				
		Enable națive code de	bugging			
	Output					
	Show output from:		- 일 일 일 점 р			
Solution Ex Property M Class view Team Explo Test Explorer	Developer PowerShe	Output Find S				
L) Keady			Ŷ	Add to Source Control 🔺 🛛 📲		

Press *Start* and the demo is running:

🌄 C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe	-	\times
Connection at Port D: no (to 0xFFFFFFF) Connection at Port B: no (to 0xFFFFFFF)		^
Connection at Port C: no (to 0xFFFFFFF)		
Line Crossed: no		
Line Crossed Flags: 0x00		
Cfg Station Address.: 0x03F4 (1012)		
PD IN Byte.Bit offset: 0.0.0 Size: 160 bits		
PD OUT Byte.Bit offset: 0.0.0 Size: 160 bits		

Slave ID 0x0000000C		
Bus Index: -65524		
Bus AutoInc Address.: 0xFFF4 (65524)		
Bus Station Address.: 0x03F5 (1013)		
Bus Alias Address: 0X03F3 (1011)		
Vendor ID	ЭН	
Product Code WX951EC00 = Unknown		
Revision		
Est Type Dect Au vog (to Av0000007)		
Connection at Port A. yes (to 0x0000000))		
Connection at Port B: no (to AVEFEFEE)		
Connection at Port C. po (to AVEFEFEE)		
Line Crossed Flags · AXA		
Cfg Station Address - 0x03E5 (1013)		
D IN Byte Bit offset: 20.0 Size: 1112 hits		
PD OUT Byte.Bit offset: 20.0.0 Size: 1016 bits		

EcMasterDemoPython runtime: 1.0s		
Press any key to continue		~



2.4.3 Visual Studio Code

Install python extension by open extension tab and enter *python*:



Open folder Examples/EcMasterDemoPython and configure the launch.json:

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Python: Aktuelle Datei",
            "type": "python",
            "request": "launch",
            "program": "${file}",
            "console": "integratedTerminal",
            "cwd": "",
            "args" : [
                "--mode", "1",
                "-f", "ENI.xml",
                "--link", "winpcap 127.0.0.1 1",
                "-b", "4000",
                "-t", "1000",
                "-v", "3",
            ],
            "env": {"PYTHONPATH": "${workspaceRoot}"}
        }
    ]
}
```





Configure linter in settings.json:



≺ .	<u>Eile E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> e	o <u>R</u> un <u>T</u> erminal <u>H</u> e	Ip settings.json - EcMasterDemoPython - Visual Studio Code —		×
G	EXPLORER		EcDemoApp.py O launch.json • O settings.json ×	° 0	
	OPEN EDITORS 1 UNSAVED ECMASTERDEMOPYTHON V.scode O launch.json O settings.json EcDemoApp.py EcDemoApp.pu EcDemoAppGui.ui EcDemoAppIni.ui	9+	<pre>.vscode > {} settingsjoon > 1 2 3 9 9 9 9 9 9 9 9 9 9 9 9 9</pre>	**************************************	T
0					
8					
563	> OUTLINE				
Pytho	n 3.7.8 64-bit 🛞 0 🛆 53 🛈 47 💡	Python: Aktuelle Datei (EcMasterDemoPython) Ln 6, Col 2 Spaces: 4 UTF-8 CRLF JSON with Comr	nents 🔗	D.

Open EcDemoApp.py and the following lines to set environment:

```
import os
import sys
INSTALLDIR = "C:/Temp/EC-Master-Windows-x86_64Bit/"
os.environ["PATH"] += os.pathsep + INSTALLDIR + "Bin/Windows/x64"
sys.path.append(INSTALLDIR + "Sources/EcWrapperPython")
sys.path.append(INSTALLDIR + "Examples/EcMasterDemoPython")
```



× <u>-</u>	<u>Eile E</u> dit <u>S</u> election <u>V</u> iew <u>o</u>	<u>Go R</u> un <u>T</u> erminal <u>H</u> elp	EcDemoApp.py - EcMasterDemoPython - Visual Studio Code —	L X
Ð	EXPLORER	💠 EcD	emoApp.py ×	▶ Ⅲ …
	> OPEN EDITORS	💠 Ect	DemoApp.py > 😫 EcLogging > 😚init	
\circ	✓ ECMASTERDEMOPYTHON	1	#/*	E. Sectors on
\sim	✓ vscode	2	# * EcDemoApp.py	Contractor Sectors
	() Jaunch ison	3	# * Copyright acontis technologies GmbH, Ravensburg, Germany	- 40°
୧୦	() launchijson	4	# * Description EC-Master demo application for Python	EDD ATTRACTOR
6	{} settings.json	5	# **/	Artan.
~	EcDemoApp.py	9+ 6	import os	
	🕏 EcDemoAppGui.py	7	import sys	Figure at
~	🗉 EcDemoAppGui.ui	8	<pre>INSTALLDIR = "C:/Temp/EC-Master-Windows-x86_64Bit/"</pre>	Particular and
~□	EcDemoAppInteractive.pv	9	os.environ["PATH"] += os.pathsep + INSTALLDIR + "Bin/Windows/x64"	ALC: NO.
Ш		10	<pre>sys.path.append(INSTALLDIR + "Sources/EcWrapperPython")</pre>	R. B. Blogarow
		11	<pre>sys.path.append(INSTALLDIR + "Examples/EcMasterDemoPython")</pre>	The second second second
		12		The Property of the Annual Party of the Part
		13	<pre>from EcWrapperPython import *</pre>	The second secon
		14	from EcWrapperPythonTypes import *	
		15	from enum import Enum	1 Mar
		16	import sys, getopt	The second secon
		17	from threading import Thread	Concernance and an and an an
		18	import threading	Contraction of the second second
		19	import time	Contraction of the local division of the loc
		20		Contraction of the second seco
		21	EC_DEMO_APP_NAME = "EcMasterDemoPython"	A State of the second s
		22		A second se
		23	class EcError(Exception):	
		24	pass	
		25		AND STREET
		26	class RunMode(Enum):	States Interest
		27	UNKNOWN = 0	Rear Source and States
_		28	Master = 1	A CONTRACTOR OF A CONTRACT
_		29	Kastilent = 2	
_		30	MDXGateWay = 3	
		31	Simulator = 4	National Control of the same of the second s
		32	def etc to is address(stuise).	TANK AND
		33	<pre>det str_to_1p_address(string): noture [int(actot) for actot in string cnlit(" ")[0.4]]</pre>	A Contraction of the second se
0		34	return [int(ottet) for ottet in string.split(".")[0:4]]	
0		35	class Feloration	NAME.
		30	def init (colf):	Contraction of the second seco
ર્જુર		37	COLE COLECTION DE CLOCHEVEL INFO	BRIDDE NAMES
	> OUTLINE	36	solf modiy = ""	
Pytho	n 3.7.8 64-bit 🛞 0 🛆 53 🛈 47	₽ Python: Aktuelle Datei (EcMaster	DemoPython) Ln 37, Col 24 Spaces: 4 UTF-8 CRLF Pyt	thon 🔊 🗘

Start debugging and the demo output will be written into the terminal:





3 FAQ

PyQt5 cannot be installed on Ubuntu 14.04 x64, because it requires Python 3.5. How can I install it?

It can be installed by calling

\$ sudo apt-get install python3-pyqt5

I installed Python and the demo crashes with strange errors. What can I do?

This might be a problem of mixing x86 with x64 binaries. Verify that if you have installed the Python runtime for x64 bit, please install also EC-Master for x64 bit.